

How did you cook it ?

About Honda's AAOS-based IVI System

Honda Motor Co., Ltd.

Electrification Business Development Operations BEV Development Center

Software Defined Mobility Development Unit

Digital Cockpit Development Division

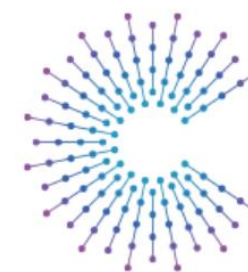
Infotainment Software Platform Development Department

Chief Architect
OSPO Tech Lead

Yuichi Kusakabe

yuichi_kusakabe@jp.honda

- 2005/4 DENSO TEN(FUJITSU TEN)
 - Renesas H8S AVN power control(Non-OS)
 - Panasonic UniPhier USB-Audio(Fastest in the industry with HDD transfer function)
- 2009/12 Working together with in Japan OEM
 - Engaged in the R&D of SoC for next-generation IVI and multimedia OS
- 2011/1 Paradigm shift to use OSS
 - Renesas RMA1 Linux-PF Project Leader
 - AGL(Automotive Grade Linux) Board Member
 - Renesas R-Car M2 original fast boot
 - Renesas R-Car H3/M3 Java、HTML5 browser
- 2020/6 Join Honda Motor in IVI software development team
 - Qualcomm SA8155P AAOS based IVI Chief Architect/OSPO Tech Lead

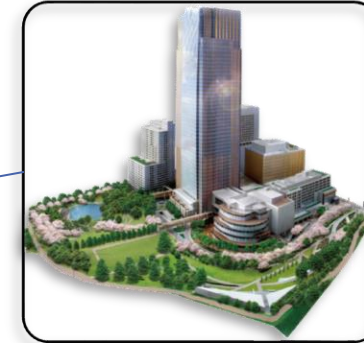


COVESA

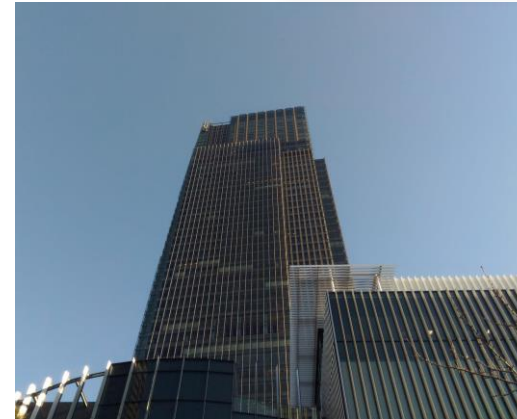
Osaka Umeda

Just
Opened!

Tokyo Roppongi



Honda
**Innovation
Lab Tokyo**



- What is IVI (in-vehicle information) system ?
- Honda infotainment History and Software scale
- Goals to aim for in IVI Software Development
- Basic concept of in-house software development
- 23M/Y US ACCORD IVI system

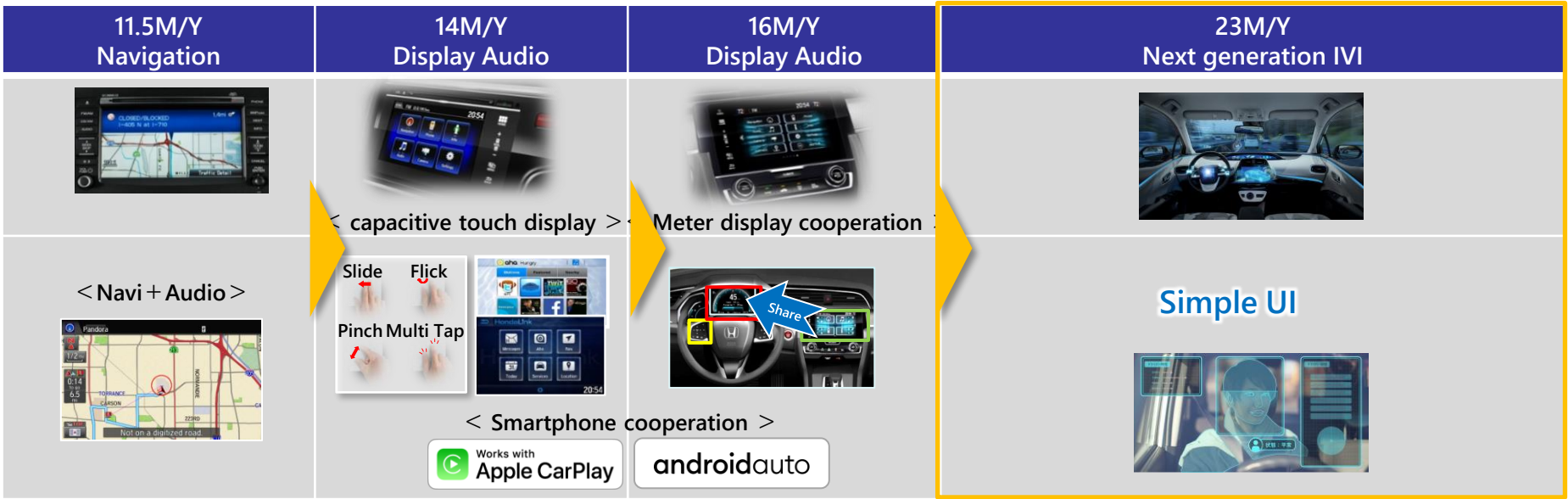
What is IVI (in-vehicle information) system ?

■ What is IVI (in-vehicle information) system ?

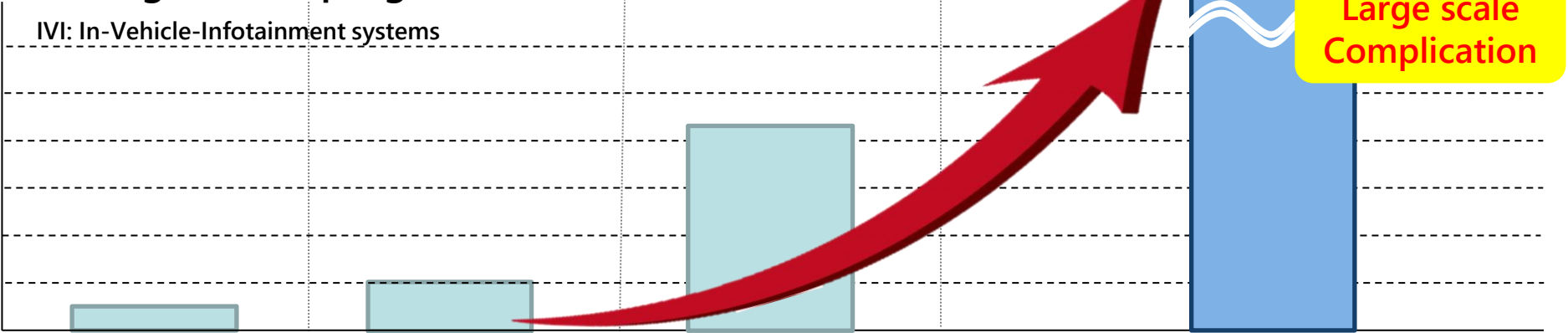
An infotainment system that connects to the outside of the vehicle, adding communication functionality



The IVI system connects various things and provides "information" and "entertainment"

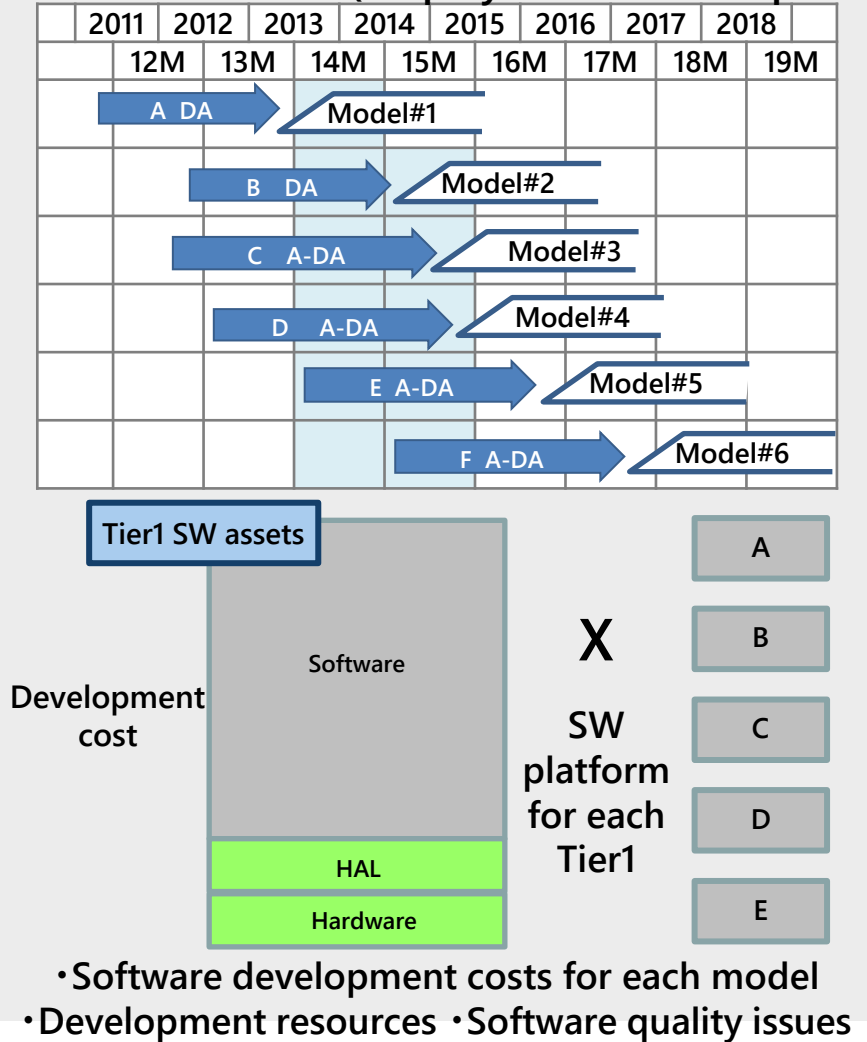


Changes in IVI program size

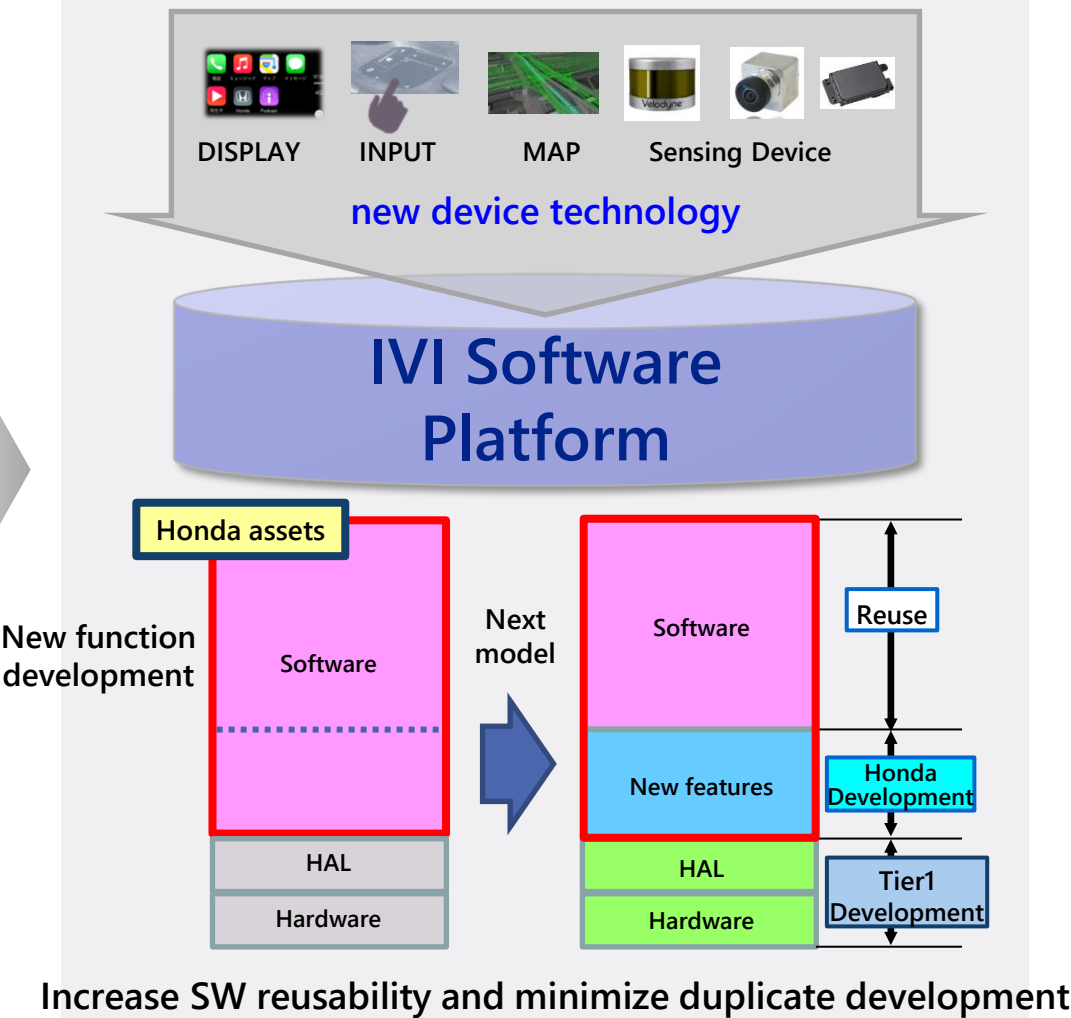


Software is the source of competitiveness, and its scale is increasing at an accelerating pace

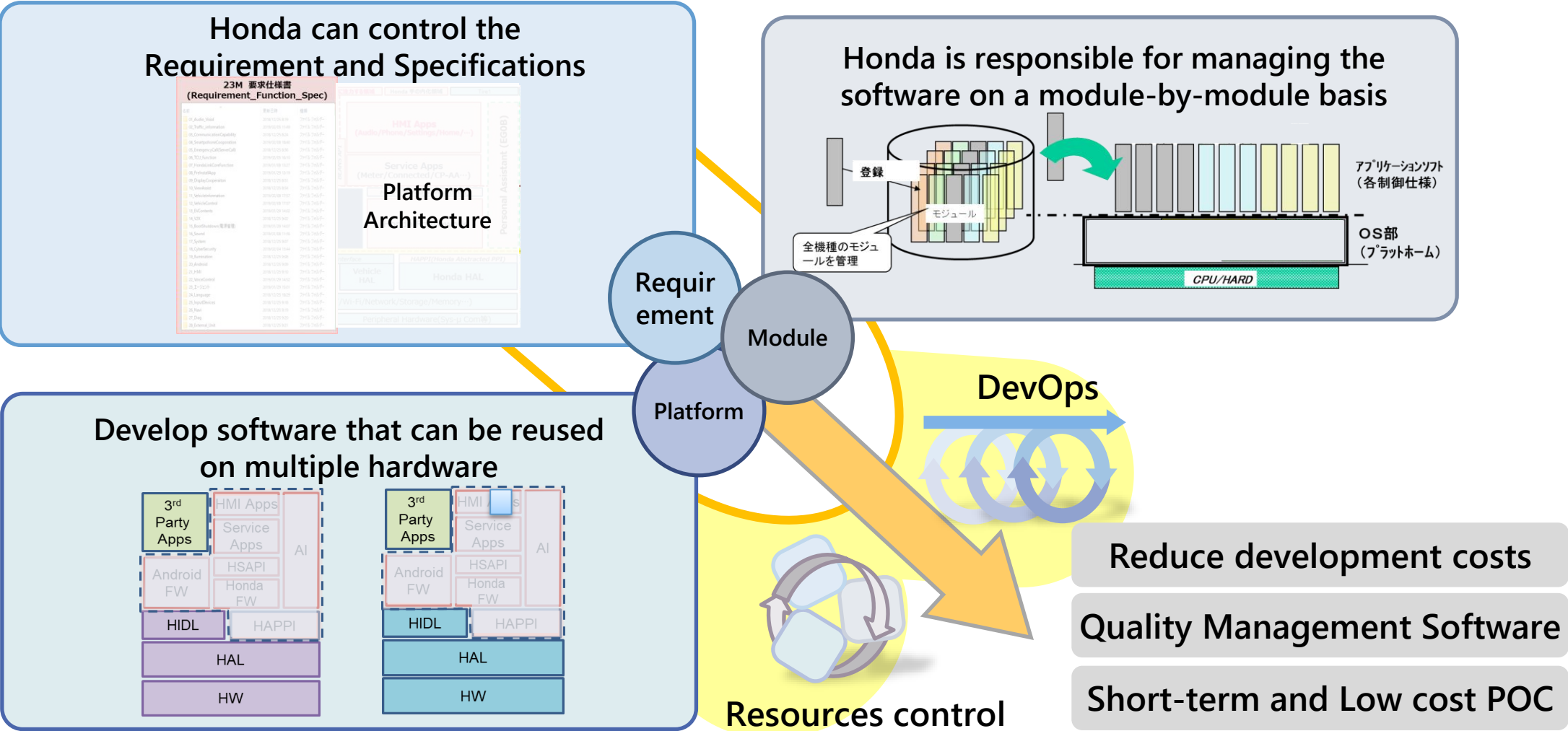
Traditional method (Display Audio development)



Software Platform construction



Aiming to improve the efficiency and quality of software development

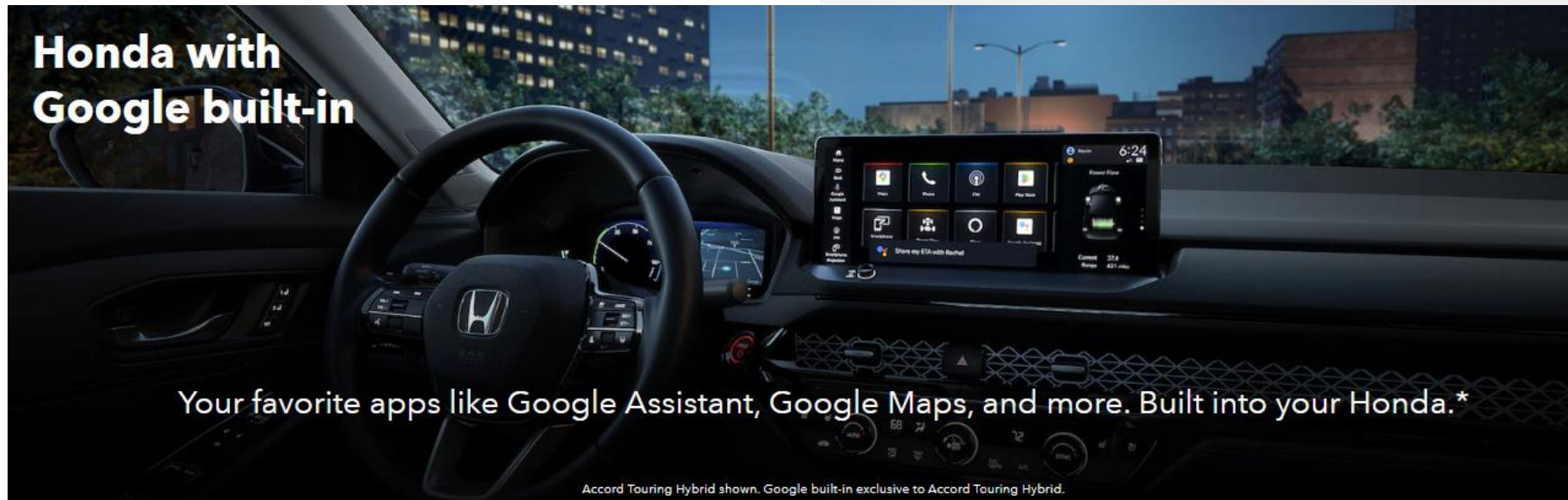


Apply to 23M/Y US ACCORD and expand to new future and model development

■ Achievement

The First-Ever Honda with Google built-in.
The 2023 Accord Touring Hybrid with Google built-in also offers an electrifying hybrid powertrain, 12-speaker Bose premium sound system, Head-Up Display, and more.

[2023 Honda Accord Google Services - YouTube](#)



IVI system will be greatly promoted on Honda.com

- Overview of in-house software development
- Project management section
- Technical section

- Overview of in-house software development
 - About Honda's DNA
 - Why does Honda implement software in-house development ?
 - Scope of in-house software development
 - Points on the division of software in-house development
 - Software development team
 - Software development process

Waigaya

"Waigaya" is Honda's unique culture of open and frank discussions about "dreams" and "ideal work styles," regardless of age or position. Instead of being venues for compromise or coordination to reach consensus, Waigaya sessions are an opportunity to create new values and concepts by thoroughly exchanging opinions in a serious and honest manner. Many of Honda's innovations, such as industry firsts and world firsts, have been born from deepening substantive discussions through Waigaya.

<https://global.honda/en/brand/hondaism/>

A00

The first thing discussed when a Honda project is launched is, "what kind of world is this work trying to realize?" One might call it a concept or guideline to ensure that we never waver until the very end. At Honda, we call this "A00." Whenever we hit a wall or have a disagreement, we always go back to it and use it as the basis for all our decisions.

| Aiming for 120% products quality

"We have to aim for 120% product quality. If 99% of the products we make are perfect, that would seem like a pretty good record. However, the customers who become the owners of the remaining 1% will surely consider their products 100% defective. It is unacceptable that even one customer in a thousand—even one customer in ten thousand—should receive a defective product. That's why we have to aim for 120%." When founder Soichiro Honda said this he defined the company's fundamental approach to quality: what it means to strive to be a company society wants to exist. Determined to meet or exceed the expectations of customers, Honda is taking new initiatives to reach ever-higher product quality standards. That is who we are.

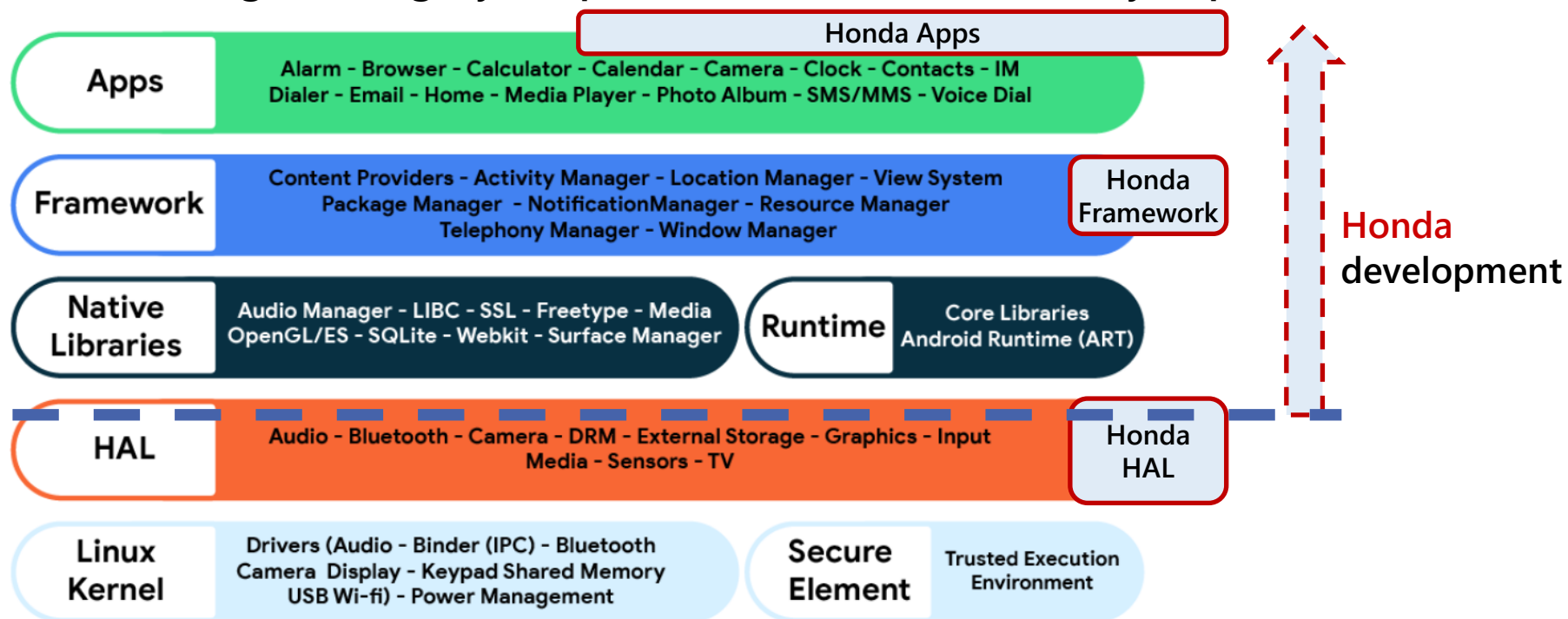
To strengthen customer trust by offering products founded in safety and offering a new level of outstanding quality, Honda has created a quality cycle that continuously enhances quality at every stage: design, development, production, sales and after-sales service.

https://global.honda/sustainability/cq_img/report/pdf/2013/report_2013-09.pdf

- Because we want to respond to the needs of our customers at an explosive speed
- It's very difficult to achieve perfect specifications from the beginning, and we want to try it out many times before delivering it to our customers.
 - “A lot of times, people don't know what they want until you show it to them. (Steve Jobs.)”
- We want to create the world's best Honda products with our own hands
- To cut off the way of excuses based on other companies, develop products in which we leading role.
- To use OSS appropriately and focus development resources on competitive areas
- We want to realize software development that can constantly change
- To acquire the skills to compete equally with engineers around the world

We are strongly promoting the project with a strong will to bring software in-house

- Honda develops software at a higher level than HAL
- Responsible for HAL and hardware at Tier 1
- Some Interfaces that are missing from the AOSP standard, Honda will design and add new Interfaces
 - Customization = Negative Legacy = Update failure, Minimize is very important



※出所 <https://source.android.com/docs/setup/about?authuser=2&%3Bhl=ja&hl=ja>

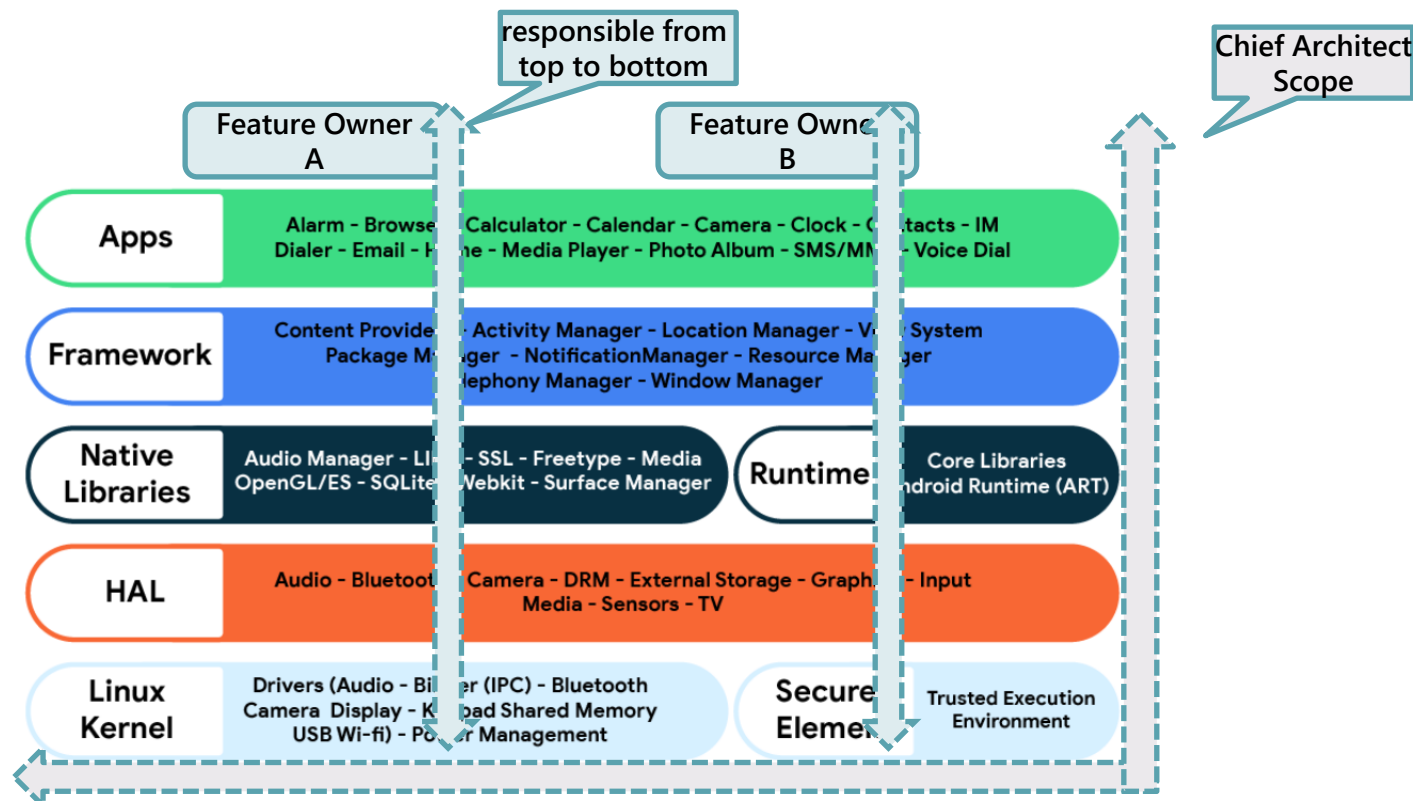
Optimize division of labor to focus on application development
Succeeded in developing a based on AAOS that in-house development of IVI software platform

- # Non-Functional requirements
-
- The diagram illustrates a stack of components for non-functional requirements. At the top is a box labeled 'App'. Below it is a horizontal line. Under the line is a box labeled 'AAudio'. Below 'AAudio' is a box labeled 'HAL'. At the bottom are two boxes, 'Mic' and 'Spk'. To the left of the stack, a red arrow points upwards from the bottom to the line, with the text 'Platform SPEC' and '60ms' in red. To the right, a red arrow points downwards from the line to the bottom, with the text 'Platform SPEC' and '35ms' in red.

1	2	3	A	B	C	D	E
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							
50							

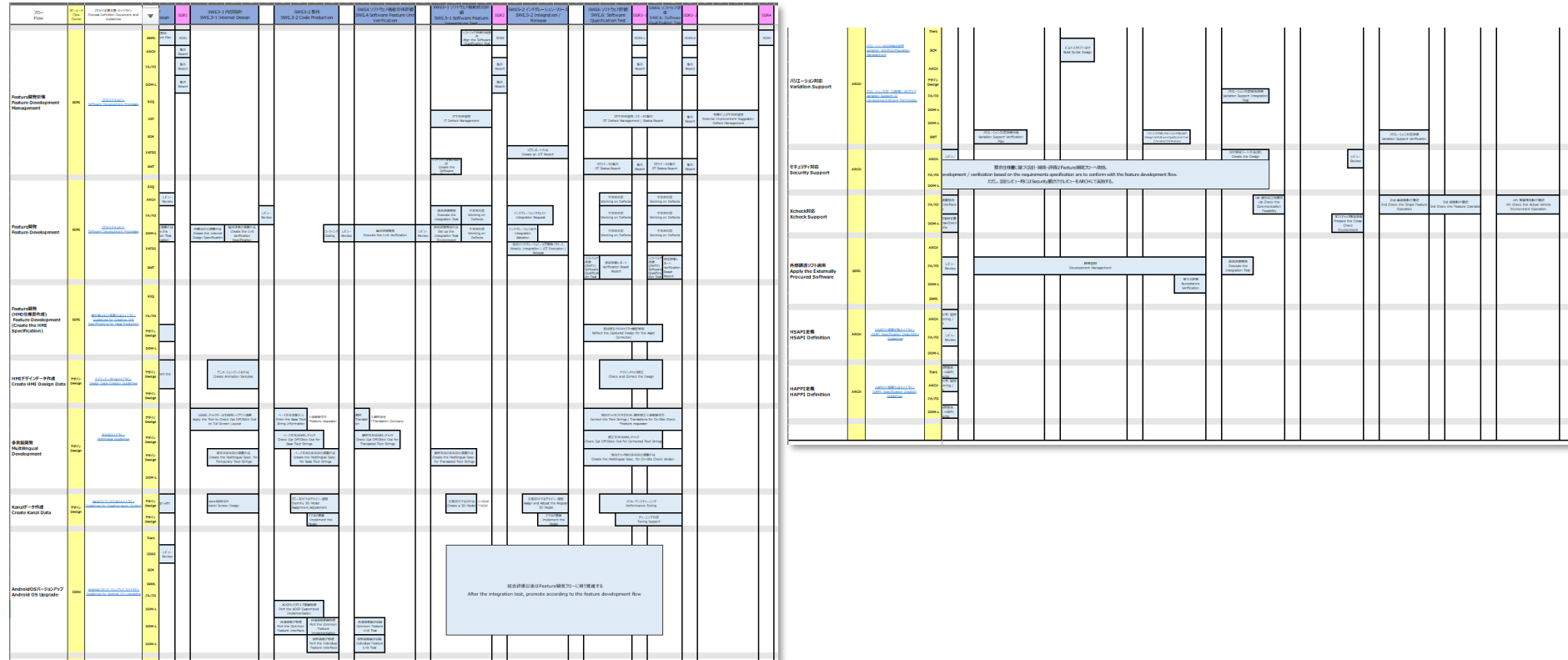
Open Source Summit Europe @Vienna 2024

- We started a software development team from scratch, everything started from scratch.
- Assign a Feature Owner to be responsible from top to bottom layers
- There are many members with different experience levels, so the position of leading the team is very important.



The Feature Owner is responsible from top to bottom and assigns each feature
Lead Architect is responsible for optimizing the All Software System

- Since the development process was also from scratch, ASPICE was a good starting point
- SEPG/SQA is independent from development team, And It checks each process.
- We have chance to Dry Run development 2 times, we were able to prepare the development process.



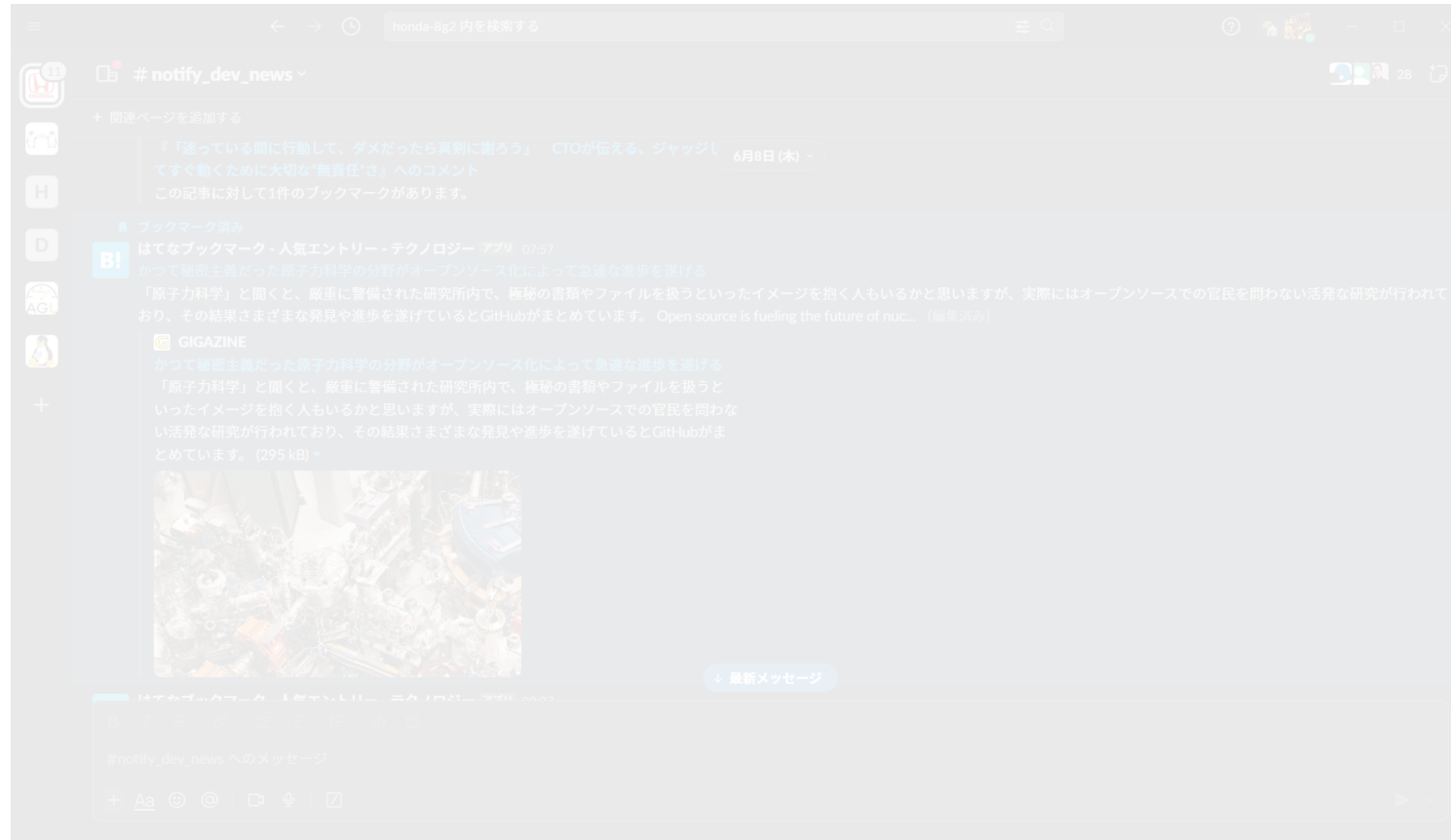
Build a development process based on ASPICE (with some exceptions for BSP)
Now that development is complete, we plan to carry out tailoring to the actual situation

■ Overview of in-house software development

■ Project management section

- Communication channels
- Many manuals (JIRA)
- Quality of source code
- About development status/progress management meeting
- Sharing of system development status
- Collaborative communication of critical issues

- Using Teams and Slack, we can communicate in real time across multiple branches
- Bad news first culture is very important
- Be careful of too many channels (Bad example: team_Feature-AxFeature-B)



We can communicate in real time across multiple locations, which is important.

- Create various rules from scratch and apply them to the development process
- Sometimes there is a mismatch with software engineers
- Rules that cannot be followed are as if there are no rules at all

JIRA manuals

- ▼ 不具合管理規約
 - ▼ 24M(PJ02DEF01)不具合チケット規約
 - ▶ PJ02DEF01不具合チケット入力ガイド
 - ▶ PJ02DEF01 環境プロファイルチケット作成ページ
 - ▶ 【DEF】 Relプランチ投入チケット（サブタスク）規約（PJ02DEF01）
 - ▶ 【不具合チケット】 横展・マル次運用ガイド
 - ▶ 不具合管理チケット規約サポート資料 - 状況別のフローチャート
- ▼ 量産開発Jiraマニュアル・規約
 - ▶ 自動ラベル付与ルール一覧
 - ▶ SDRチケットマニュアル
 - ▶ 周知依頼事項チケットマニュアル
 - ▶ ソフト要求仕様書の要求QAチケットマニュアル
 - ▶ feature開発レビューチケットマニュアル
 - ▶ TE打ち上げチケットマニュアル
 - ▶ feature1チケットマニュアル
 - ▶ ソフトfeatureチケットマニュアル
 - ▶ アウトプットチケットマニュアル
 - ▶ サブタスクチケットマニュアル
 - ▶ 制限事項チケットマニュアル
 - ▶ プロセスチケットマニュアル
 - ▶ フォルダチケットマニュアル
 - ▶ タスクチケットマニュアル
 - ▶ 仕様確認質問チケットマニュアル
 - ▶ リスク課題プロジェクトマニュアル
 - ▶ featureプロジェクトマニュアル
 - ▶ 開発運用プロジェクトマニュアル

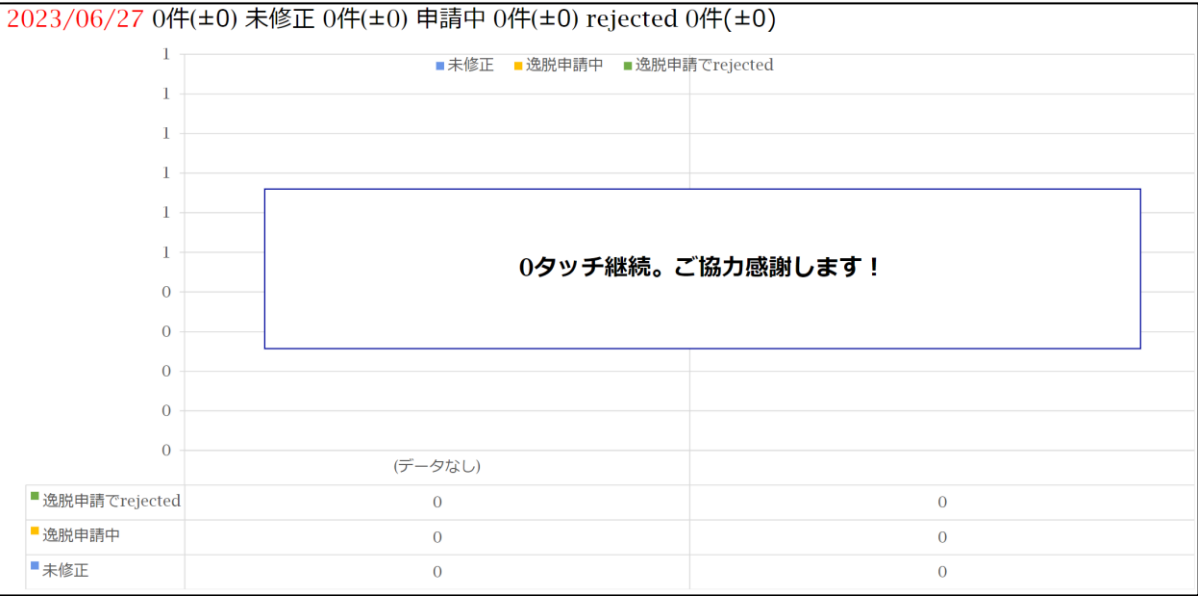
Issue JIRA manual

- 目次
- 更新履歴
- イントロダクション
 - 規約の上位の文書
 - 参照資料
 - 関連資料
 - 略号・用語
- 本書について
- 対象読者
- 不具合管理チケットの起票対象
- 本書の構成
- ステータス
 - ステータスの説明
- 不具合管理チケットのワークフロー
 - 役割
 - ワークフロー
- インクリメンタルプロセスによる追加施策
 - インクリメンタルプロセスにおける優先度
 - 優先度の選択肢と意味
- 登録の規約
 - 登録時のルール・手順
 - ルール
 - 手順
 - 起票方法
 - 起票時の注意
 - チケット項目
 - 起票（ステータス：Draft）
 - 起票元デグレ情報の定義
 - 着手待ち（ステータス：Draft→着手待ち）
- 改修の規約
 - 調査時のルール・手順
 - ルール
 - 手順
 - チケット項目

The manuals were created from scratch, and some of the rules were mismatched with the actual site
We will review the rules in the future and make them easy for anyone to use

- Adopt industry standard coding rules for source code
- Monitor the number of NGs using various static analysis tools

対象	分類	規約内容	検出方法と対策	適用する成果物
Java	スタイル	グーグルのコーディング規約（AOSP Java Code Style for Contributors）に準拠すること。	<ul style="list-style-type: none">コミット時に自動チェックする（AndroidStudio用の自動チェックツール"CheckStyle")<ul style="list-style-type: none">AOSP_Java_コーディング規約自動チェック不可の項目はレビューで確認する(実施要否はRASIC上の承認者が判断)指摘に対する対応基準として自身が作成した箇所の指摘は全て修正する。	ソースコード
	命名規則	上記コーディング規約に従う		
	規約からの変更点 ※背景	<ul style="list-style-type: none">コードの各行の長さは、最大200文字へ変更"Order import statements"(インポートステートメントの順序)において"各グループ内でアルファベット順に並べる"は削除変数定義から使用するまでの行数を3→7へ変更（7行以内はOK、8行以上はNG）HONDA拡張<ul style="list-style-type: none">AOSPで定義されているIntentを暗黙的に使用してstartActivityを起動しないこと（理由：旧：AOSP標準アプリの対応(Ω以前)）		
	使用禁止API	下記のAPIは使用禁止とする <ul style="list-style-type: none">Java.util.Timer	<ul style="list-style-type: none">CIの使用禁止APIチェックツールを用いて検出。結果はJIRAチケット化してDEFにてフォロー	
	セキュアコーディング（※背景） ▲VER1.10 [2021/09/23]	<p>CERTのコーディングスタンダードに準拠すること（下記を参照）</p> <ul style="list-style-type: none">JPCERT/CC翻訳：Androidアプリケーション開発へのルール適用 <p>JSSECのセキュリティ設計ガイドラインを参考にする</p> <ul style="list-style-type: none">Android アプリのセキュア設計・セキュアコーディングガイド	<ul style="list-style-type: none">静的解析ツールのCoverityを使用し、違反が無いかをチェックする。尚、静的解析ツールが検出可能なCERTルールチェックは下記を参照<ul style="list-style-type: none">Coverityのバージョン毎の情報	



Don't use originals, use de facto standard coding rules

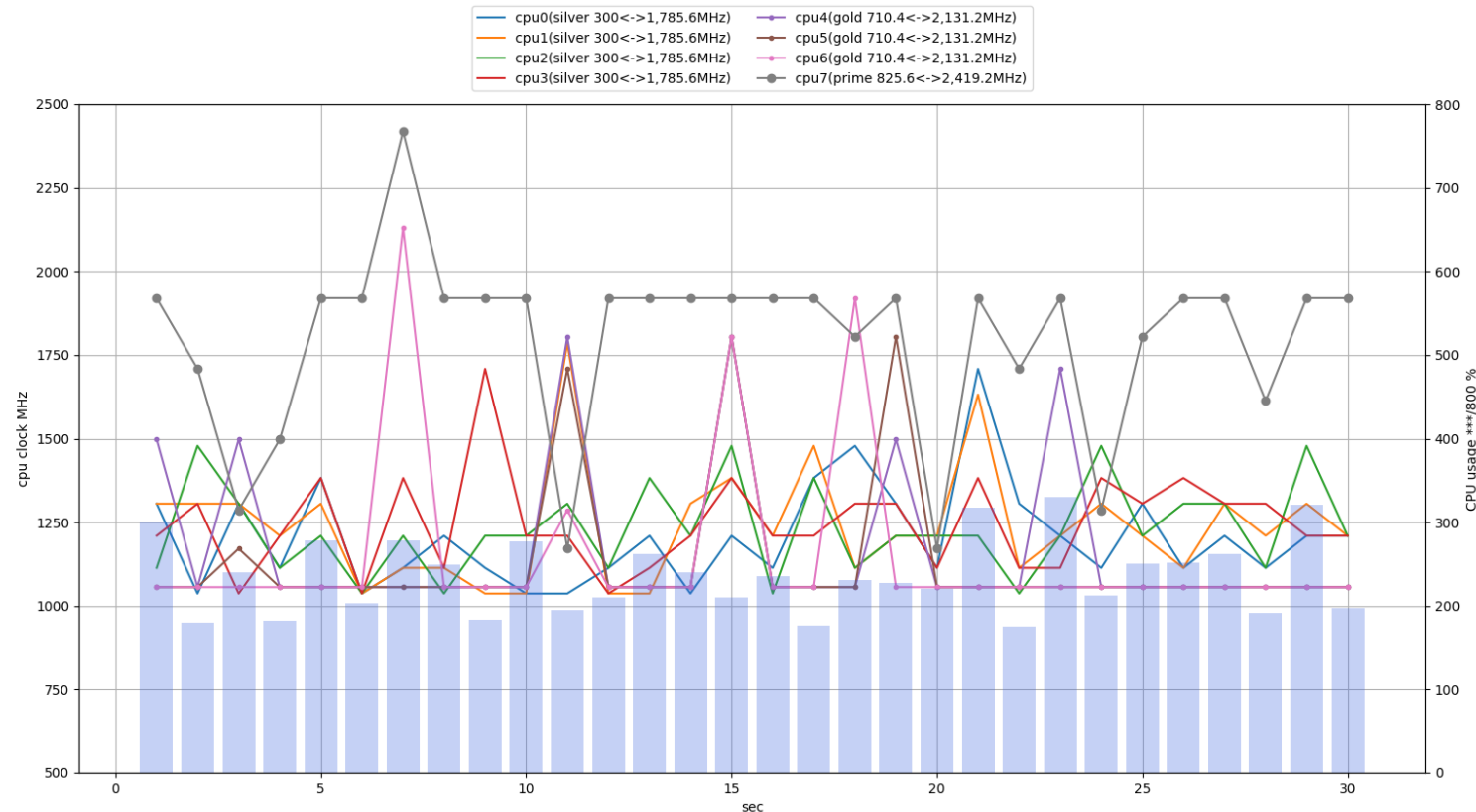
Automate tools to support consistent quality levels

- Simple meeting schedule, no unnecessary meetings
- OEM internal meetings: process status, management risks
- General development meeting: Feature Owners , PM and any person report each status
- Features meeting: Information sharing meeting within each Feature group

Minimize meeting schedules and focus on product source code

- We hold a general meeting where all developers gather every week to directly share information
- Broken telephony sometimes occurs when there are many developers

general meeting information : CPU load



Everyone can share their voices through meetings where everyone can participate

- For long-term or critical issues, make the status of JIRA easily visible to everyone
- Speed up resolution with Lead Architect taking the lead
- For particularly important issues, report directly to the project manager and share risks

JIRA Critical issues

本ページについて

FG/ISP限らずICB全体の中で技術的なフォローが必要な超重要案件について、本ページにてフォローを実施する

キー	要約	T	作成	更新	期限	担当者	報告者	P	ステータス	解決
PJ02DEF01-3004	【PJ01DEF01-15924横展】 【GLICB-2876】 【System UI】 BTA再生中、BTA再生画面にてTEL MuteスイッチをON/OFF後に、MCSの[DefaultAssistant]ボタンを押下してGoogle Assistantを起動したとき、フリーズすることがあった	🔴	2023/02/10	2023/05/08		APN Miyako Enel	APN Yuichi Fukui	=	終了	完了
PJ02DEF01-350	【PJ01DEF01-7420横展】 CLONE - [Qcom問い合わせ]LaneWatch 映像・ガイドラインがピクピク動くときがある	🔴	2022/09/09	2023/01/30		APN Tomoaki Sato(Sound)	APN Yo Takahashi	^	終了	完了
PJ02DEF01-61	[toA12]Output SELinux trace derived from Qualcomm base software	🔴	2022/08/24	2022/11/18		APN Yuichi Fukui	APN Yuichi Fukui	=	終了	完了
PJ01RSK00-1683	[技術課題: CarPlay][MFI認証][Cinemo側]RTD要件達成できていない: Cinemo→iPhoneへ送信遅延	🟡	2021/12/09	2022/03/07		BRS Shuuji Matsuhisa	HGT Yuichi Kusakabe	^	終了	完了
PJ01RSK00-1544	[技術課題] Gaia Assitant Echo Reference 取得方法の確認	🟡	2021/08/06	2021/09/21		HGT Yuichi Kusakabe	APN Yo Takahashi	=	解決済み	対応済み
PJ01RSK00-1540	RPCのCPU負荷削減について	🟡	2021/08/05	2021/08/06		HGT Yuichi Kusakabe	HGT Yuichi Kusakabe	=	解決済み	対応済み
PJ01RSK00-1536	[技術課題: CarPlay][MFI認証][TestID:9F82D] WiFi Latency 改善 (パラつき改善)	🟡	2021/07/30	2022/12/13		APN Yoshihiro Takamura	APN Yoshihiro Takamura	^	解決済み	完了

For important issues, the key point is whether to assign good engineers
Just following the schedule will not solve the problem

- Overview of in-house software development
- Project management part
- Technical section
 - Continuous access to information
 - Update to the new AAOS version
 - About vulnerability response
 - About adding complex features
 - Point of Native implementation
 - About GKI (Generic Kernel Image) topics
 - About software development with hardware spec
 - About use OSS

- Various information is published on websites
- it is necessary to always check the information

Manage boot time

The boot process is a chain of actions starting from the boot ROM, followed by the bootloader, kernel, **Init**, **Zygote**, and **system server** (bold indicates Android-specific boot process). In the automotive-specific boot process, early services such as rearview camera (RVC) must start during the kernel boot.

Order	Component	Android	Android Automotive
1	Boot ROM	Loads the first stage of the boot loader into internal RAM.	
2	Bootloader	Initiates memories, verifies for security, and loads kernel.	
3	Kernel	Sets up interrupt controllers, memory protection, caches, and scheduling; launches user space processes.	Rearview camera (RVC) process starts early in kernel boot. After the process is up, GPIO from VMCU triggers RVC to show on display.
4	Init process	Parses <code>init.rc</code> scripts, mounts file systems, launches Zygote, launches system process.	Vehicle HAL (VHAL) starts during init phase as part of core services. Can be moved to earlier states depending on ServiceManager; requires removing dependencies on shared libraries (such as init).
5	Zygote	Sets up Java Runtime and init memory for Android objects.	
6	System server	First Java component in the system, starts core Android services.	CarService starts after all system services are started.

<https://source.android.com/?hl=ja&authuser=2>
<https://developer.android.com/?hl=ja>
<https://source.android.com/docs/setup/about/site-updates?hl=ja&authuser=2&version=13>

Android Automotive 14 release details

This page summarizes new major features provided in Android Automotive 14.

Features

Car framework

Display and Window Manager

1. Concurrent multi-user `SysUI` and Lock screen.
2. Rear-seat input lock.
3. Special input handling.
4. Support rear-seat input lock and the sharing of Special input key handling for concurrent multi-users in single AAOS instances.
5. Display and task mirroring. Support Display and Task Mirroring for concurrent multi-users in a single AAOS instance.
6. Multi-user IME. Support a multi-user IME for concurrent multi-users in a single AAOS instance.
7. Enable `TaskView` on a secondary home or passenger display.

It is important to watch many information resources, share them with all developers

- Android software is released regularly, and features may be added in addition to bug fixes
 - QPR1, QPR2, QPR3
- However, due to unique customization and fragmentation, updates are difficult
- If Android customization is required to implement old specifications, review the specifications and reconsider feasibility from a software perspective many times.

	Android11	Android12	Android13
GPP	'20/09/02	'21/10/05	'22/08/15
QPR1	'20/12/07	'22/01/15	'22/12/05
QPR2	'21/03/01	'22/02/09	'23/02/01
QPR3	'21/06/17	'22/03/09	'23/04/20

<https://android.googlesource.com/platform/manifest/>
https://source.android.com/docs/devices/automotive/start/releases/s_qpr3_release?hl=ja&authuser=2

Android Automotive 12 QPR3 release details

This page summarizes new features provided in Android Automotive OS 12 QPR3.

Product details

System UI and core apps improvements

To support the launch of YouTube in cars, updates were made to:

- Settings > ClosedCaptioning
- Settings > TalkBack

**In order to constantly update software, it is very important to review past requirements
Honda reviewed Requirements to achieve in-house software production**

- Regular security updates required after SOP
 - In the case of Kernel LTS updates, cooperation from SoC vendors may also be required
- The key point is how quickly you can incorporate the security patches that are released every month.

Kernel LTS

The following table describes the minimum kernel version requirement for Security Patch Level compliance. The Android Launch Version refers to the version of the Android OS that was on the device at launch and the kernel version refers to the version of the Linux kernel that is presently on the device.

★ **Note:** Devices that launched on Android 11 & 12 with the 5.4 kernel are required to update these devices to the 5.4.147 kernel (or higher) for compliance with the 2022-05-05 SPL.

References	Android OS Launch Version	Kernel Version	Minimum Update Version
A-202441831	11	5.4	5.4.147
A-204345773	12	5.4	5.4.147

<https://source.android.com/docs/security/bulletin/2022-05-01?hl=ja#kernel-lts>

Minimize customization to ensure regular security patch updates



Photo taken by Bong Grit/Getty Images

BILBAO, Spain: At the Open Source Summit Europe, Jonathan Corbet, Linux kernel developer and executive editor of Linux Weekly News, caught everyone up with what's new in the Linux kernel and where it's going from here.

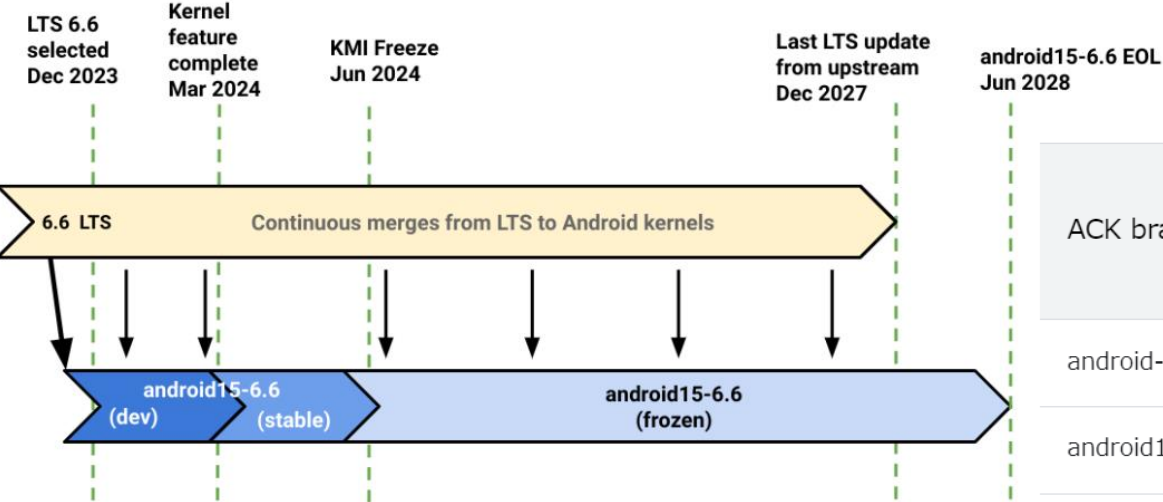
Here's one major change coming down the road: Long-term support (LTS) for Linux kernels is being reduced from six to two years.

Longterm release kernels

Version	Maintainer	Released	Projected EOL
6.6	Greg Kroah-Hartman & Sasha Levin	2023-10-29	Dec, 2026
6.1	Greg Kroah-Hartman & Sasha Levin	2022-12-11	Dec, 2026
5.15	Greg Kroah-Hartman & Sasha Levin	2021-10-31	Dec, 2026
5.10	Greg Kroah-Hartman & Sasha Levin	2020-12-13	Dec, 2026
5.4	Greg Kroah-Hartman & Sasha Levin	2019-11-24	Dec, 2025
4.19	Greg Kroah-Hartman & Sasha Levin	2018-10-22	Dec, 2024

<https://www.kernel.org/category/releases.html>

https://www.zdnet.com/article/long-term-support-for-linux-kernel-to-be-cut-as-maintenance-remains-under-strain/#google_vignette



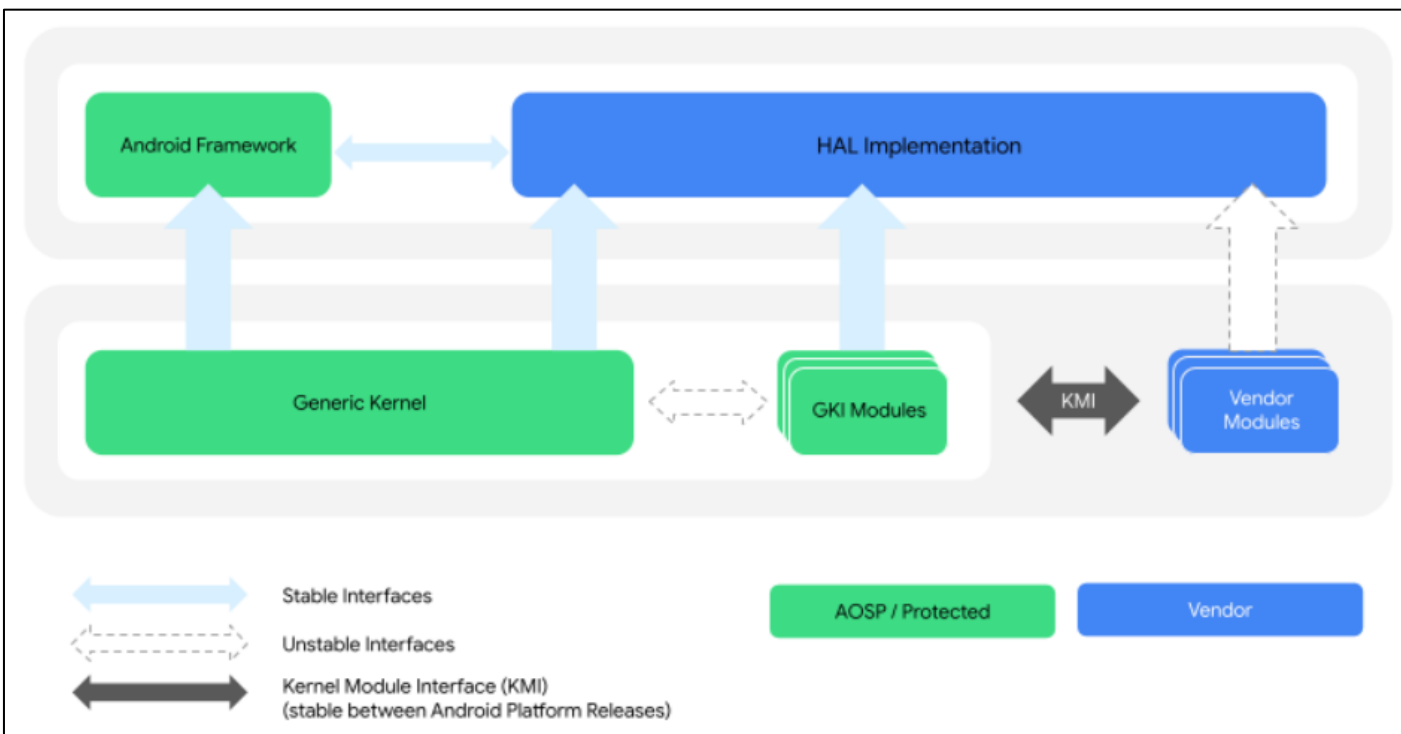
ACK branch	Launch date	Support lifetime (years)	EOL
android-4.19-stable	2018-10-22	6	2025-01-01
android11-5.4	2019-11-24	6	2026-01-01
android12-5.4	2019-11-24	6	2026-01-01
android12-5.10	2020-12-13	6	2027-07-01
android13-5.10	2020-12-13	6	2027-07-01
android13-5.15	2021-10-31	6	2028-07-01
android14-5.15	2021-10-31	6	2028-07-01
android14-6.1	2022-12-11	6	2029-07-01
android15-6.6	2023-10-29	4	2028-07-01

Longterm release kernels

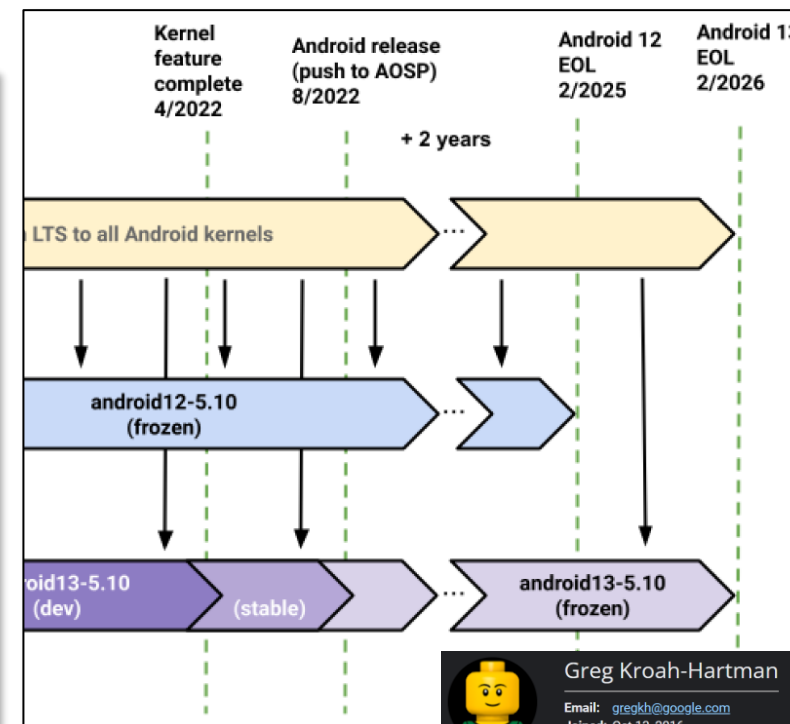
Version	Maintainer	Released	Projected EOL
6.6	Greg Kroah-Hartman & Sasha Levin	2023-10-29	Dec, 2026
6.1	Greg Kroah-Hartman & Sasha Levin	2022-12-11	Dec, 2026
5.15	Greg Kroah-Hartman & Sasha Levin	2021-10-31	Dec, 2026
5.10	Greg Kroah-Hartman & Sasha Levin	2020-12-13	Dec, 2026
5.4	Greg Kroah-Hartman & Sasha Levin	2019-11-24	Dec, 2025
4.19	Greg Kroah-Hartman & Sasha Levin	2018-10-22	Dec, 2024

<https://source.android.com/docs/core/architecture/kernel/android-common>
<https://www.kernel.org/category/releases.html>

- Preventing fragmented kernels customized by each company
- Kernel binary image is provided by Google (configs/gki_defconfig)
- When customizing Kernel, support for upstream or Vendor Module is required
- KMI fix timing is fast, so it is necessary to take action



<https://source.android.com/docs/core/architecture/kernel/generic-kernel-image?hl=ja>
<https://android-review.googlesource.com/q/project:kernel/common+branch:android13-5.15>



Greg Kroah-Hartman
Email: gregkh@google.com
Joined: Oct 13, 2016

Subject	Owner
Revert "gpiolib: Fix irq_domain resource tracking for gpiochip_irqchip_add_domain()"	Greg Kroah-Hartman
Revert "net: Remove DECnet leftovers from flow.h."	Greg Kroah-Hartman
Revert "net: Remove DECnet leftovers from flow.h."	Greg Kroah-Hartman

There is a Kernel Fix event before the Android release, so you need to react quickly

- In order to add new Features, consideration is also given to the FW side (HOTWORD etc.)
- If you want to create Low-Latency audio functionality, AAudio is available.

Sharing audio input

Audio input usually comes from the built-in mic, an external mic, or an audio interface attached to the device. Audio input can also come from a phone conversation.

Sometimes two or more apps might both want to "capture" the same audio input. They may be performing different tasks. For example, some apps that receive audio might be "recording," like a simple voice recorder, while other apps might be "listening," like the Google Assistant or an accessibility service that respond to voice commands.

In either case, these apps want to receive audio input. Throughout this page, we use the term "capture" regardless of whether an app is recording or just listening.

If two or more apps want to capture audio at the same time, there can be a problem delivering the audio signal from the same source to all of them. This page describes how the Android system shares audio input between multiple apps that capture audio.

<https://developer.android.com/guide/topics/media/sharing-audio-input?hl=ja>
<https://developer.android.com/ndk/guides/audio/aaudio/aaudio?hl=ja>

AAudio

★ **Note:** Developers should consider using the open source Oboe library which is available on [GitHub](#). Oboe is a C++ wrapper that provides an API that closely resembles AAudio. It calls AAudio when it is available, and falls back to [OpenSL ES](#) if AAudio is not available.

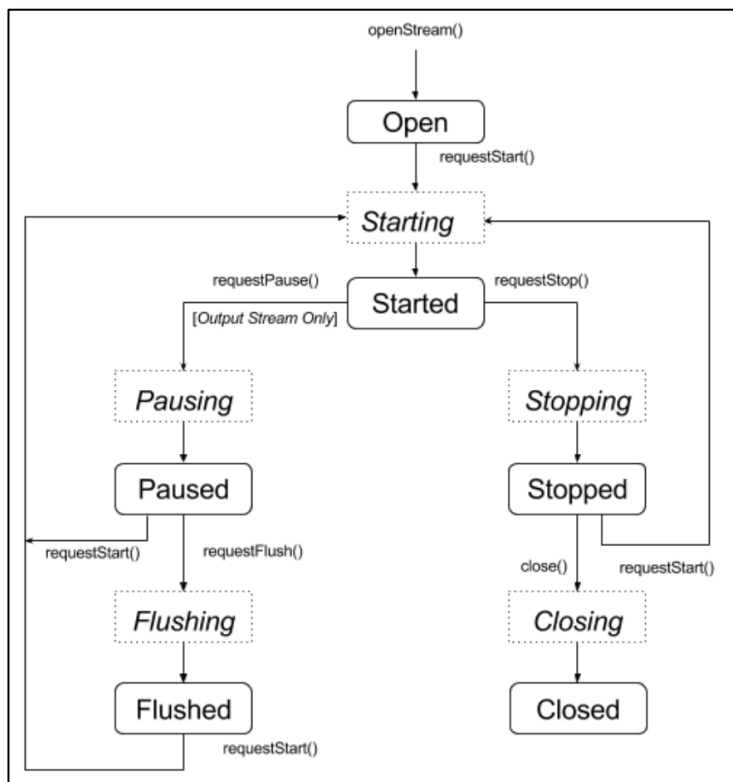
AAudio is a new Android C API introduced in the Android O release. It is designed for high-performance audio applications that require low latency. Apps communicate with AAudio by reading and writing data to streams.

The AAudio API is minimal by design, it doesn't perform these functions:

- Audio device enumeration
- Automated routing between audio endpoints
- File I/O
- Decoding of compressed audio
- Automatic presentation of all input/streams in a single callback.

Req new feature, it is important to use existing modules instead of creating your own

- When using Native such as AAudio, a perfect understanding of the documentation is required
- Hardware knowledge is also required, such as state transitions, how to set up buffers, non-thread safety, etc.



※出所 <https://developer.android.com/ndk/guides/audio/aaudio/aaudio?hl=ja>

Thread safety

The AAudio API is not completely **thread safe**. You cannot call some of the AAudio functions concurrently from more than one thread at a time. This is because AAudio avoids using mutexes, which can cause thread preemption and glitches.

To be safe, don't call `AAudioStream_waitForStateChange()` or read or write to the same stream from two different threads. Similarly, don't close a stream in one thread while reading or writing to it in another thread.

Calls that return stream settings, like `AAudioStream_getSampleRate()` and `AAudioStream_getChannelCount()`, are thread safe.

These calls are also thread safe:

- `AAudio_convert*ToText()`
- `AAudio_createStreamBuilder()`
- `AAudioStream_get*()` except for `AAudioStream_getTimestamp()`

★ **Note:** When a stream uses a callback function, it's safe to read/write from the callback thread while also closing the stream from the thread in which it is running.

Basically, it is important to read and implement what is written in the document
Low-level layers also require knowledge of hardware

- Some devices have a limited lifespans and the hardware SPEC must be considered
- Even if the app writes 1 byte, 1 byte is not written to the device
- It is important to understand how caching works after an app writes data

Flash wear management

Android Automotive internal storage uses an Embedded MultiMediaCard (eMMC) with thousands of erase/write cycles; if the eMMC fails, the system can become unusable. As vehicles have long lifespans (typically 10+ years), the eMMC must be extremely reliable. This page describes eMMC behavior and how OEMs can lower the risk of a failing eMMC (and thus avoid failed Android Automotive systems).

eMMC behavior

eMMC devices use wear leveling techniques to work around erase/write limitations by arranging data and distributing writes evenly across the system (so no single block fails due to intensive writes). The estimated life of eMMC depends on:

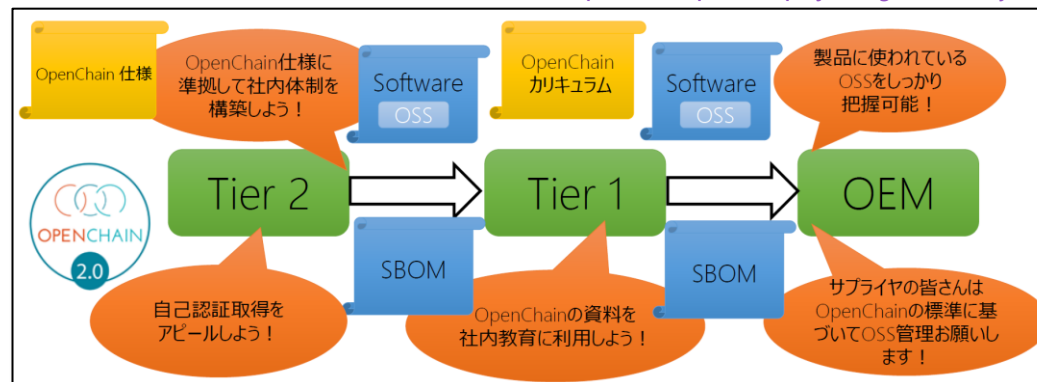
- **Amount of writes.** On phones, the amount of data written to internal storage can be more than 10 GB per day. On Automotive implementations, we don't have real world data on how much data will be written due to limited apps. However, when users are streaming high quality music and using navigation, we observe 50 MB data written to eMMC every minute. In the future, we may have other types of write-intensive apps, such as dashboard camera apps that continuously record and store videos. In addition, some cars are shared vehicles used multiple hours everyday. For these reasons and others, we expect Android Automotive implementations to have more eMMC writes than a phone.
- **Write patterns.** Writes and erasures are done in blocks. Writing data frequently in small chunks wears

<https://source.android.com/docs/core/perf/flash-wear?hl=ja>

It is necessary to design software that takes hardware SPEC into consideration
Android also has a mechanism to monitor the data of writes to storage

- OSS is essential for product development, and the use of OSS complies with OpenChain
- We adopted OpenChain's SPDX Lite (Excel) to smoothly share OSS information between companies.

<https://www.openchainproject.org/community-of-conformance>



SPDX Liteの項目例			-パッケージダウンロード位置 (入手先)		-ライセンスへのコメント	
-パッケージ名	-パッケージSPDX識別子	-パッケージバージョン	-解析したファイル (手作業の場合false)	-ホームページ (開発コミュニティサイト)	-著作権テキスト	-変更の有無
-パッケージファイル名	-パッケージバージョン	-パッケージファイル名	-結論されたライセンス	-宣言されたライセンス	-パッケージに関するコメント	-ライセンス識別子
SPDXサンプルの一部抜粋						
3.1	3.2	3.3	3.4	3.7	3.8	3.11
PackageName	Package SPDX Identifier	Package Version	PackageFileName	PackageDownloadLocation	Files Analyzed	PackageHomePage
3.13	3.15					
Concluded License	Declared License					
MIT	MIT	LibXML2	libxml2-2.9.9.tar.gz	http://xmlsoft.org/libxml2/	FALSE	http://xmlsoft.org/
BSD-3-Clause	BSD-3-Clause	PCRE	pcre-8.43.tar.gz	http://ftp.pcre.org/pub/pcre/	FALSE	https://www.pcre.org/
その他(ライセンス名を記載)	その他(ライセンス名を記載)	SQLite	sqlite-autoconf-3300100.tar.gz	https://www.sqlite.org/download.html	FALSE	https://www.sqlite.org/index.html
Zlib	Zlib	Zlib (1.2.3)	zlib-1.2.11.tar.gz	https://www.zlib.net/	FALSE	https://www.zlib.net/
MIT	MIT	cURL (7.41.0)	curl-7.66.0.tar.bz2	https://github.com/curl/curl/releases	FALSE	https://curl.haxx.se/

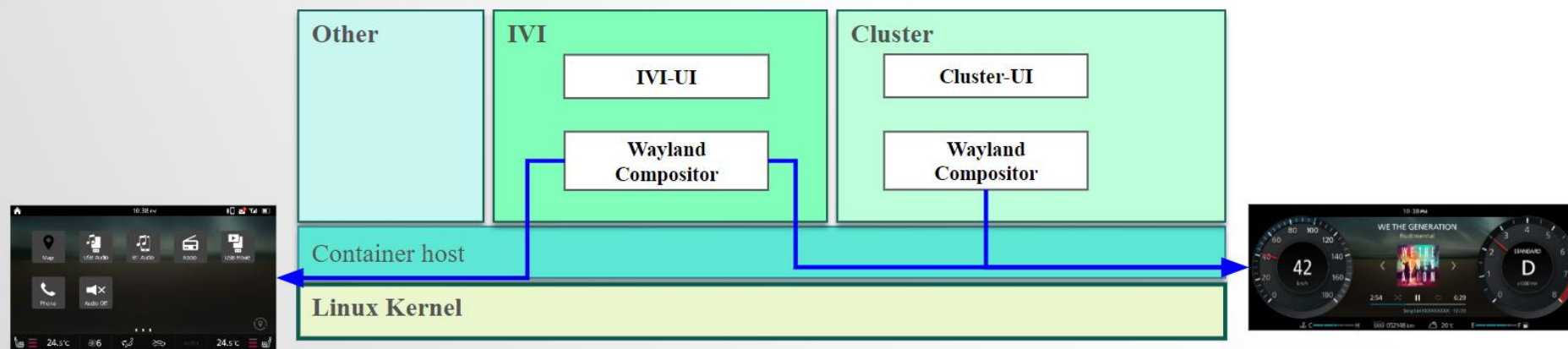
https://www.meti.go.jp/shingikai/mono_info_service/sangyo_cyber/wg_seido/wg_bunyaodan/software/pdf/003_04_00.pdf
<https://github.com/OpenChain-Project/Japan-WG-General/tree/master/License-Info-Exchange>



We obtained OpenChain self-certification in December last year
This year Honda is launched OSPO(Open Source Program Office)

Technical detail for QM Isolation

- Display isolation and sharing.
 - Cluster and IVI have separate requirement.
 - Cluster UI require to high frame rate.
 - IVI UI is difficult to control drawing timing for each IVI apps.
 - QM isolation selected separate compositor architecture.
 - It realize to isolate composite and frame update timing between cluster and IVI.



- With “in-house software development” as our important keyword, we have completed the development of Honda's first IVI software platform and are now able to launch ACCORD in North America
- However, this is just the first step, and using the software platform we have created, Honda is moving to the next stage
- There are many challenges to overcome in order to realize SDV, such as software development speed, automation, and contribution to the OSS community
- There is no limit to the potential of external communities such as the OSS community
- We look forward to your continued collaboration with our engineers as we move toward the next stage that Honda aims to reach

Let's boost the Automotive industry together

HONDA
The Power of Dreams

How we move you.
CREATE ► TRANSCEND, AUGMENT